

## IN THE CLAIMS

Please amend the claims as indicated:

1. (currently amended) A method for replacing a cryptology key in a computer module, wherein said computer module includes a plurality of evictable cryptology keys, said method comprising:

determining, for each of a plurality of evictable cryptology keys in a computer module, a replacement expense for each said evictable cryptology key, said replacement expense determined by:

a probability that each said evictable cryptology key will be needed by the computer module after said evictable cryptology key is evicted, and

an amount of cycle time required to re-store, if evicted, each said evictable cryptology key in the computer module, wherein said cycle time is determined by calculating a number of generations to a nearest ancestor cryptology key that is required to unwrap said least expensive evictable cryptology key, said nearest ancestor cryptology key being from a plurality of non-evicted remaining cryptology keys in the computer module;

identifying a least expensive evictable cryptology key based on said replacement expense; and

replacing said least expensive evictable cryptology key with a replacement cryptology key.

2. (original) The method of claim 1, said step of replacing said least expensive cryptology key further comprising:

locating a blob comprising said least expensive evictable cryptology key and a security software shell;

removing said security software shell from said blob; and

storing said least expensive evictable cryptology key in said computer module.

3. (currently amended) The method of claim 1, wherein an expense to re-load an evictable cryptology key is determined by both an expense to reload a child evictable cryptology key as well as an expense to re-load any ancestor cryptology keys of the child evictable cryptology key.  
~~further comprising:~~

~~determining said cycle time by calculating a number of generations to a nearest ancestor of said least expensive evictable cryptology key, said nearest ancestor being from a plurality of non-evicted remaining cryptology keys in the computer module.~~

4. (original) The method of claim 3 further comprising:

storing, if a parent cryptology key of said least expensive evictable cryptology key is not stored in said computer module, a child cryptology key of said nearest ancestor key of said least expensive evictable cryptology key; and

repeating said storing step until said least expensive evictable cryptology key is stored in said computer module.

5. (original) The method of claim 1, wherein the computer module is a Trusted Platform Module (TPM).

6. (currently amended) A data-processing system capable of replacing a cryptology key in a computer module, wherein said computer module includes a plurality of evictable cryptology keys, said data-processing system comprising:

means for determining, for each of a plurality of evictable cryptology keys in a computer module, a replacement expense for each said evictable cryptology key, said replacement expense determined by:

a probability that each said evictable cryptology key will be needed by the computer module after said evictable cryptology key is evicted, and

amount of cycle time required to re-store, if evicted, each said evictable cryptology key in the computer module, wherein said cycle time is determined by calculating a number of generations to a nearest ancestor cryptology key that is required to unwrap said least expensive evictable cryptology key, said nearest ancestor cryptology

key being from a plurality of non-evicted remaining cryptology keys in the computer module;

means for identifying a least expensive evictable cryptology key based on said replacement expense; and

means for replacing said least expensive evictable cryptology key with a replacement cryptology key.

7. (original) The data processing system of claim 6, said means for replacing said least expensive cryptology key further comprising:

means for locating a blob comprising said least expensive evictable cryptology key and a security software shell;

means for removing said security software shell from said blob; and

means for storing said least expensive evictable cryptology key in said computer module.

8. (currently amended) The data processing system of claim 6, wherein an expense to re-load an evictable cryptology key is determined by both an expense to reload a child evictable cryptology key as well as an expense to re-load any ancestor cryptology keys of the child evictable cryptology key. ~~further comprising:~~

~~means for determining said cycle time by calculating a number of generations to a nearest ancestor of said least expensive evictable cryptology key, said nearest ancestor being from a plurality of non-evicted remaining cryptology keys in the computer module.~~

9. (original) The data processing system of claim 8 further comprising:

means for storing, if a parent cryptology key of said least expensive evictable cryptology key is not stored in said computer module, a child cryptology key of said nearest ancestor key of said least expensive evictable cryptology key; and

means for repeating said storing step until said least expensive evictable cryptology key is stored in said computer module.

10. (original) The data processing system of claim 6, wherein the computer module is a Trusted Platform Module (TPM).

11. (currently amended) A computer usable medium for replacing a cryptology key in a computer module, wherein said computer module includes a plurality of evictable cryptology keys, said computer usable medium comprising:

computer program code for determining, for each of a plurality of evictable cryptology keys in a computer module, a replacement expense for each said evictable cryptology key, said replacement expense determined by:

a probability that each said evictable cryptology key will be needed by the computer module after said evictable cryptology key is evicted, and

an amount of cycle time required to re-store, if evicted, each said evictable cryptology key in the computer module, wherein said cycle time is determined by calculating a number of generations to a nearest ancestor cryptology key that is required to unwrap said least expensive evictable cryptology key, said nearest ancestor cryptology key being from a plurality of non-evicted remaining cryptology keys in the computer module;

computer program code for identifying a least expensive evictable cryptology key based on said replacement expense; and

computer program code for replacing said least expensive evictable cryptology key with a replacement cryptology key.

12. (original) The computer usable medium of claim 11, said computer program code for replacing said least expensive cryptology key further comprising:

computer program code for locating a blob comprising said least expensive evictable cryptology key and a security software shell;

computer program code for removing said security software shell from said blob; and

computer program code storing said least expensive evictable cryptology key in said computer module.

13. (currently amended) The computer usable medium of claim 11, wherein an expense to re-load an evictable cryptology key is determined by both an expense to reload a child evictable

cryptology key as well as an expense to re-load any ancestor cryptology keys of the child evictable cryptology key. ~~further comprising:~~

~~computer program code for determining said cycle time by calculating a number of generations to a nearest ancestor of said least expensive evictable cryptology key, said nearest ancestor being from a plurality of non-evicted remaining cryptology keys in the computer module.~~

14. (original) The computer usable medium of claim 13 further comprising:

computer program code for storing, if a parent cryptology key of said least expensive evictable cryptology key is not stored in said computer module, a child cryptology key of said nearest ancestor key of said least expensive evictable cryptology key; and

computer program code for repeating said storing step until said least expensive evictable cryptology key is stored in said computer module.

15. (original) The computer usable medium of claim 11, wherein the computer module is a Trusted Platform Module (TPM).

16. (new) The method of claim 3, further comprising:

prior to evicting a parent cryptology key, determining how many child cryptology keys of the parent cryptology key will be disabled by the evicting of the parent cryptology key; and

evicting from a plurality of parent cryptology keys a parent cryptology key that has been determined to affect fewer child cryptology keys than other parent cryptology keys in the plurality of parent cryptology keys.

17. (new) The data processing system of claim 6, further comprising:

means for, prior to evicting a parent cryptology key, determining how many child cryptology keys of the parent cryptology key will be disabled by the evicting of the parent cryptology key; and

means for evicting from a plurality of parent cryptology keys a parent cryptology key that has been determined to affect fewer child cryptology keys than other parent cryptology keys in the plurality of parent cryptology keys.

18. (new) The computer usable medium of claim 11 further comprising:

computer program code for, prior to evicting a parent cryptology key, determining how many child cryptology keys of the parent cryptology key will be disabled by the evicting of the parent cryptology key; and

evicting from a plurality of parent cryptology keys a parent cryptology key that has been determined to affect fewer child cryptology keys than other parent cryptology keys in the plurality of parent cryptology keys.